*A Project report*

*on*

**FACIAL RECOGNITION BASED ATTENDANCE SYSTEM**

*submitted in partial fulfillment of the requirement for the award of degree of*

**BACHELOR OF TECHNOLOGY**

*In*

**COMPUTER SCIENCE & ENGINEERING**

*By*

MOHAMMAD REHAN ALI 18JJ5A0507

NARSINGOJU HEMANTH 18JJ5A0510

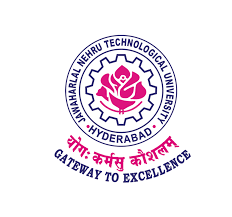
SHYAMANTHULA SAIPAVAN 18JJ5A0512

GUNDU SAIKRISHNA 16JJ1A0524

*Under the guidance of*

***Dr.D.Ramesh***

*Professor of CSE*



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**JAWAHARLALNEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD COLLEGE OF ENGINEERING & TECHNOLOGY**

Nachupally , Kondagattu , Jagtial Dist. – 505501, T.S.

**2020**

**JAWAHARLALNEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD COLLEGE OF ENGINEERING & TECHNOLOGY**

Nachupally , Kondagattu , Jagtial Dist. – 505501, T.S.

**2020**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**CERTIFICATE**

This is to certify that the mini project entitled “**FACIAL RECOGNITION BASED ATTENDANCE SYSTEM** ” is being submitted by

MOHAMMAD REHAN ALI 18JJ5A0507

NARSINGOJU HEMANTH 18JJ5A0510

SHYAMANTHULA SAIPAVAN 18JJ5A0512

GUNDU SAI KRISHNA 16JJ1A0524

in partial fulfillment of the requirements for the award of **BACHELOR OF TECHNOLOGY** to **JNTU, Hyderabad**. This record is a Bonafide work carried out by them under my guidanceand supervision. The result embodied in this project report has not been submitted to any otheruniversity or institute for the award of any degree.

**Internal Guide External Guide**

**-------------------- --------------------**

**Head of The Department.**

**--------------------------------**

**ACKNOWLEDGEMENT**

When a Task gets Successful, we feel happy.

But that happiness is nothing when we don’t state the persons who supported us

to make it a success.

Success is not a phase we enjoy more, it’s the journey that we enjoy more. So, the people who helped us in our journey should be acknowledged.

We came across the 4 years of beautiful journey in this college and this project completes it. We would like to express deep gratitude and respect to all those people behind the screen who guided, inspired and helped us for the completion of our project work.

This is a very good project and this made a progress to all our team mate’s career.

I would like to express my thankfulness to my project guide, **Dr. D. RAMESH,** Professor of CSE for his constant motivation and valuable help through the project work, and I express my gratitude to **Dr. T. VENUGOPAL**, Professor & Head, Department of Computer Science and Engineering for giving us the opportunity to carry out his project work.

We are very much grateful to our project coordinator **Dr. T. VENUGOPAL**, Professor of CSE for his constant mentoring and cooperation during the project.

We show gratitude to our beloved principal **Dr. N.V. RAMANA** and Vice Principal **Dr.** **S. VISWANADHA RAJU** for providing necessary infrastructure and resources for theaccomplishment of our project report at JNTUHCEJ

I also extend my thanks to my Team Members for their co-operation during my course. Finally, I would like to thank my friends for their co-operation to complete this project.

**CONTENTS**

**NAME** PGNO

1. ABSTRACT………………………………………...................1
2. INTRODUCTION……………………………………………...2
3. EXISTING SYSTEM ………………………………............. 3-6
4. FEASIBILITY STUDY.......................................................... 7-8
5. PROPOSED SYSTEM........................................................ 9-10
6. ALGORITHM………………………….................................. 11-13
7. HARDWARE AND SOFTWARE REQUIREMENTS.......... 14
8. APPROACH…………………………………………………...15
9. WORKING……………………………………………………. 16
10. TOOLS USED........................................................................ 17
11. DESCRIPTION OF PROJECT………………………………..18
12. SYSTEM TESTING………………………………………...…42
13. APPLICATIONS………………………………………………46
14. FUTURE SCOPE………………………………………………47
15. CONCLUSION…………………………………………………47
16. REFERENCES............................................................................48

**ABSTRACT:**

To maintain the attendance record with day to day activities is a challenging task. The conventional method of calling the name of each student is time consuming and there is always a chance of proxy attendance. The following system is based on face recognition to maintain the attendance record of students. The daily attendance of students is recorded subject wise which is stored already by the administrator. As the time for corresponding subject arrives the system automatically starts taking snaps and then apply face detection and recognition technique to the given image and the recognized students are marked as present and their attendance update with corresponding time and subject id. We have used deep learning techniques to develop this system, histogram of oriented gradient method is used to detect faces in images and deep learning method is used to compute and compare facial features of students to recognize them. Our system is capable to identify multiple faces in real time.

Educational institutes are facing major security issues these days. Consequently, they need several specially trained personnel to attain the desired security. These personnel, as human beings, make mistakes normally but a trained machine can’t. A proposed solution to the aforementioned matter is a Facial Recognition Based Security System, which can detect intruders to restricted or high-security areas, and help in minimizing human error. This system is composed of two parts: hardware part and software part. The hardware part consists of a camera, while the software part consists of face-detection and face-recognition algorithms software. When a person enters to the zone in question, a series of snapshots are taken by the camera and sent to the software to be analyzed and compared with an existing database of trusted people.

**INTRODUCTION**

Real time face recognition is part of the field of biometrics. Biometrics is the ability for a computer to recognize a human through a unique physical trait. Face recognition provides the capability for the computer to recognize a human by facial characteristics. Today, biometrics is one of the fastest growing fields in advanced technology. Predictions indicate a biometrics explosion in the next century, to authenticate identities and avoid and unauthorized access to networks, database and facilities.

A facial recognition device is a device that takes an image or a video of a human face and compares it to another image faces in a database. The structure, shape and proportions of the faces are compared during the face recognition steps. In addition, distance between the eyes, nose, mouth and jaw, upper outlines of the eye sockets, the sides of the mouth, location of the nose and eyes, and the area surrounding the check bones are also compared.

**EXISTING SYSTEM**

Many face recognition software has been implemented during the past decade. Each software uses different methods and different algorithm than other software. Some facial recognition software extracts the face features from the input image to identify the face. Other algorithms normalize a set of face images and then compress the face data, the saves the data in one image that can be used for facial recognition. The input image is compared with the face data. New method for face recognition is being used which is the three-dimensional facial recognition. In this method, a 3-D sensor is used to capture information about the shape of the face so that only distinctive features of the face, such as the contour of eye sockets, nose and chin, are used for face recognition. This new method offers some advantages over other algorithms in that recognition it is not affected by the change of light, and the face can be identified from a variety of angles, including profile view. Another new technique in facial recognition is called skin texture analysis. This technique uses the visual details of the skin, as captured in standard digital or scanned images and then turns the unique lines, patterns, and spots apparent in a person’s skin into a mathematical space. Below is an introduction for some of the existing facial recognition programs that were used for security reasons

**Face First:** Face First is a software that provides a fully automated, user friendly, turnkey mobile and live-video surveillance facial recognition system. This software generates an alert whenever a face is recognized and this occurs when the match of the input face with a face in the database is above a user defined probability. The advantage of Face First system is the availability to work in low resolution environments enabling real-world performance.

**Morpho Trak:** Morpho Trak provides biometric and identity management solutions to a broad array of markets including law enforcement, border control, driver licenses, civil identification, and facility/IT security. Morpho Trak is part of the world’s largest biometric company and leading innovator in large fingerprint identification systems, facial and iris recognition, as well as secure credentials.

**Cross Match Technologies:** Cross Match Technologies is a leading global provider of biometric identity management systems, applications and enabling technologies to governments, law enforcement agencies and businesses around the world. Offerings include biometric technologies capable of wireless, mobile or stationary use that encompass facial recognition systems and other systems.

**FEASIBILITY STUDY**

**Project Description**: The proposed solution is a real-time face recognition system that reads a video from a camera connected to the computer running the software, detects any face present in front of the camera, and then checks if this face is present in a set of face images in a database using face recognition technique.

Here we are trying to develop a system to mark attendance automatically by using image processing technique. An efficient face recognition algorithm has to be developed which can recognize person’s efficiently and also for image processing we have to have effective platform to test our algorithm.

**Possible Solutions**:

In this project we have implemented the automated attendance system. We have projected our ideas to implement “Automated Attendance System

Based on Facial Recognition”, in which it imbibes large applications. The application includes face identification, which saves time and eliminates chances of proxy attendance because of the face authorization. Hence, this system can be implemented in a field where attendance plays an important role.

It is not intrusive, can be done from a faraway distance even without the person being aware that he/ she is being scanned.

· It will be useful in banks or government offices.

· It can be used for surveillance purposes like searching for wanted criminals, suspected terrorists, missing children, for old age people.

**Most Feasible Solution:**

Nowadays Educational institutions are concerned about regularity of student attendance. This is mainly due to students’ overall academic performance is affected by his or her attendance in the institute. Mainly there are two conventional methods of marking attendance which are calling out the roll call or by taking student sign on paper. They both were more time consuming and difficult. Hence, there is a requirement of computer-based student attendance management system which will assist the faculty for maintaining attendance record automatically**.**

**Conclusion:**

In this system we have implemented an attendance system for a lecture, section or laboratory by which lecturer or teaching assistant canrecord students’ attendance. It saves time and effort, especially if it is a lecture with huge number of students. Automated Attendance System has been envisioned for the purpose of reducing the drawbacks in the traditional (manual) system. This attendance system demonstrates the use of image processing techniques in classroom. This system can not only merely help in the attendance system, but also improve the goodwill of an institution.

**PROPOSED SYSTEM**

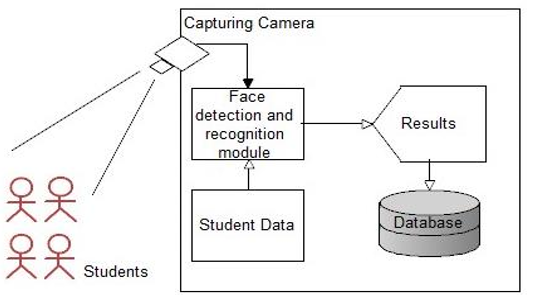
The proposed solution is a real-time face recognition system that reads a video from a camera connected to the computer running the software, detects any face present in front of the camera, and then checks if this face is present in a set of face images in a database using face recognition technique.

Here we are trying to develop a system to mark attendance automatically by using image processing technique. An efficient face recognition algorithm has to be developed which can recognize person’s efficiently and also for image processing we have to have effective platform to test our algorithm.

The software is divided into two parts: face detection and face recognition.

A proposed method to the problem in the safety and Security System, which can detect intrudes to restricted or high security areas with help of face recognition system.

The present system of attendance marking i. e, manually calling out the roll call by the faculty have quite satisfactorily served the purpose. With the change in the educational system with the introduction of new technologies in class room such as virtual class room, the traditional way of taking attendance may not be viable anymore. Even with rising number of course of study offered by universities, processing of attendance manually could be time consuming. Hence, in our project we aim at creating a system to make attendance using facial recognition technology in classrooms and creating an efficient database to record them.

**Blockdiagram:** 

**Figure: Proposed system**

The block diagram in figure describes the proposed system for Face Recognition based Classroom attendance system. The system requires a camera installed in the classroom at a position where it could capture all the students in the classroom and thus capture their images effectively. This image is processed to get the desired results. The working is explained in brief below:

**• Capturing Camera**: Camera is installed in a classroom to capture the face of the student. The camera has to be places such that it captures the face of all the

students effectively. This camera has to be interfaced to computer system for further processing either through a wired or a wireless network. In our prototype we use the in-built camera of the laptop.

• **Image Processing**: Facial recognition algorithm is applied on the captured image. The image is cropped and stored for processing. The module recognizes the images of the students face which have been registered manually with their names and ID codes in the database. The whole process requires the following steps:

a) **Train Database**: Initially we take facial image of the enrolled students. In our system we have taken three images each. This data is used later used in the facial recognition algorithm. All the cropped image of the face is resized.

b) **Face Detection and cropping**: The captured image of the classroom is initially scanned to detect faces. The detected faces are cropped and resized , same as the train database.

c) **Face Recognition**: For recognition the feature locations are refined and the face is normalized with eyes and month in fixed locations. Images from the face tracker are used to train a frontal Eigen space, and the leading three eigenvectors are retained. Since the face images have been warped into frontal views a single eigen space is enough. Face recognition

is then performed using the Eigen face approach with additional temporal information added. The projection coefficients of all images of each person are modelled as a Gaussian distribution and the face is classified based on the probability of match.

d) **Attendance Recording**: We use Excel spreadsheet to store the recorded attendance for easy-to-use output format, which is also the software which is familiar to majority of the institution staffs. This is done using Spreadsheet Link EX toolbox. If a student is recognized, the corresponding cell is updated with ‘1’, else a ‘0’. Using the formatting in the Excel, we can effectively retrieve the information effectively.

**Algorithm Suggested for Face Detection**

**SUPPORT VECTOR MACHINE**

In this Project, Supervised ML is used, especially classification tasks are done by using Support Vector Machine Classifier. The basics of Support Vector Machines and how it works are best understood with a simple example. Let’s imagine we have two tags: red and blue, and our data has two features: x and y. We want a classifier that, given a pair of (x, y) coordinates, outputs if it’s either red or blue. We plot our already labelled training data on a plane: Our labelled data A support vector machine takes these data points and outputs the hyperplane (which in two dimensions it’s simply a line) that best separates the tags. This line is the decision boundary: anything that falls to one side of it we will classify as blue, and anything that falls to the other as red. In 2D, the best hyperplane is simply a line But what exactly is the best hyperplane? For SVM, it’s the one that maximizes the margins from both tags. In other words: the hyperplane (remember it’s a line in this case) whose distance to the nearest element of each tag is the largest. Not all hyperplanes are created equal Nonlinear data Now this example was easy, since clearly the data was linearly separable — we could draw a straight line to see.

The kernel trick: In our example we found a way to classify nonlinear data by cleverly mapping our space to a higher dimension. However, it turns out that calculating this transformation can get pretty computationally expensive: there can be a lot of new dimensions, each one of them possibly involving a complicated calculation. Doing this for every vector in the dataset can be a lot of work, so it’d be great if we could find a cheaper solution. Here’s a trick: SVM doesn’t need the actual vectors to work its magic, it actually can get by only with the dot products between them. This means that we can sidestep the expensive calculations of the new dimensions! This is what we do instead:

• Imagine the new space we want: z = x² + y² 40

• Figure out what the dot product in that space looks like: a · b = xa · xb + ya · yb + za · zb a · b = xa · xb + ya · yb + (xa² + ya²) · (xb² + yb²)

• Tell SVM to do its thing, but using the new dot product — we call this a kernel function. That’s it! That’s the kernel trick, which allows us to sidestep a lot of expensive calculations. Normally, the kernel is linear, and we get a linear classifier. However, by using a nonlinear kernel (like above) we can get a nonlinear classifier without transforming the data at all: we only change the dot product to that of the space that we want and SVM will happily chug along. Note that the kernel trick isn’t actually part of SVM. It can be used with other linear classifiers such as logistic regression. A support vector machine only takes care of finding the decision boundary.

**DEEP LEARNING**

Deep Learning is a branch of Machine Learning which provides different methodologies to solve classification & regression problems of Supervised ML. In deep learning, a computer model learns to perform classification tasks directly from images, text, or sound. Deep learning models can achieve state-of-the-art accuracy, sometimes exceeding human-level performance. Models are trained by using a large set of labelled data and neural network architectures that contain many layers. Deep Learning is the most exciting and powerful branch of Machine Learning. It's a technique that teaches computers to do what comes naturally to humans: learn by example

⚫ Deep Learning models can be used for a variety of complex tasks:

⚫ Artificial Neural Networks (ANN) for Regression and classification

⚫ Convolution Neural Networks (CNN) for Computer Vision.

⚫ Recurrent Neural Networks (RNN) for Time Series analysis.

⚫ Self-organizing maps for Feature extraction.

⚫ Deep Boltzmann machines for Recommendation systems.

⚫ Auto Encoders for Recommendation systems.

**CONVOLUTIONAL NEURAL NETWORKS**

In neural networks, Convolutional neural network (ConvNets or CNNs) is one of the main categories to do images recognition, images classifications. Object detections, recognition faces etc., are some of the areas where CNNs are widely used. CNN image classifications take an input image, process it and classify it under certain categories (E.g., Dog, Cat, Tiger, Lion). Computers sees an input image as array of pixels and it depends on the image resolution. Based on the image resolution, it will see h x w x d (h = Height, w = Width, d = Dimension). E.g., An image of 6 x 6 x 3 array of matrix of RGB (3 refers to RGB values) and an image of 4 x 4 x 1 array of matrix of grayscale image. Technically, deep learning CNN models to train and test, each input image will pass it through a series of convolution layers with filters (Kernels), Pooling, fully connected layers (FC) and apply SoftMax function to classify an object with probabilistic values between 0 and 1. The below figure is a complete flow of CNN to process an input image and classifies the objects based on values. Neural network with many convolutional layers Convolution Layer Convolution is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel. 42 Figure 5.13: Image matrix multiplies kernel or filter matrix Consider a 5 x 5 whose image pixel values are 0, 1 and filter matrix 3 x 3. Image matrix multiplies kernel or filter matrix Then the convolution of 5 x 5 image matrix multiplies with 3 x 3 filter matrix which is called “Feature Map” as output shown in below Figure 5.15: 3 x 3 Output matrix 43 Convolution of an image with different filters can perform operations such as edge detection, blur and sharpen by applying

filters. Strides Stride is the number of pixels shifts over the input matrix. When the stride is 1 then we move the filters to 1 pixel at a time. When the stride is 2 then we move the filters to 2 pixels at a time and so on. The below figure shows convolution would work with a stride of 2. Stride of 2 pixels Padding Sometimes filter does not fit perfectly fit the input image. We have two options:

• Pad the picture with zeros (zero-padding) so that it fits

• Drop the part of the image where the filter did not fit. This is called valid padding which keeps only valid part of the image. Non-Linearity (ReLU) ReLU stands for Rectified Linear Unit for a non-linear operation. The output is ƒ(x) = max(0, x). Why ReLU is important:

ReLU’s purpose is to introduce non-linearity in our ConvNet. Since, the real-world data would want our ConvNet to learn would be non-negative linear values. 44 Figure 5.17: ReLU operation There are other non-linear functions such as tanh or sigmoid that can also be used instead of ReLU. Most of the data scientists use ReLU since performance wise ReLU is better than the other two. Pooling Layer Pooling layers section would reduce the number of parameters when the images are too large. Spatial pooling also called subsampling or down sampling which reduces the dimensionality of each map but retains important information. Spatial pooling can be of different types: • Max Pooling • Average Pooling • Sum Pooling Max pooling takes the largest element from the rectified feature map. Taking the largest element could also take the average pooling. Sum of all elements in the feature

**How to implement SVM in python?**

In python, scikit learn is a widely used library for implementing machine learning algorithms is available in scikit learn library and follow structure

1.Import library,

2.object creation,

3.fitting model and

4.prediction

**Pros:**

* It works really well with clear margin of separation.
* It is effective in high dimensional spaces.
* It is effective in case where number of dimensions is greater than the number of samples.

**Software Requirements**:

* Operating System: Windows
* Language: Python3
* Library: Open CV, Pandas

**Hardware Requirements**:

* Processor: Intel Core i3 or i5
* RAM: 2 or 3 GB
* Hard disk: Minimum 1GB
* Web camera

**APPROACH**

There are many difficulties related to human facial recognition. The fact that human faces are all relatively similar, yet produce varying facial expressions makes it more difficult to generalize an algorithm. Except in the case of identical twins, the face is arguably a person’s most unique physical characteristic. Each face has certain distinguishable facial features. These are the peaks and the valleys that make up the different facial features. Lighting conditions and the angle from which the facial image is taken are other factors to consider. Taking all this into account, it is important to note that humans themselves can distinguish a multitude of different faces quickly and with high accuracy. The facial recognition software is based on the ability to first recognize faces, which are a technological feat in itself, and then measure the various features of each face.

The planned testing approach is to have a database of numerous faces that is used to test the recognition algorithm against certain particular faces. The variables that need to be tested for 0 a face against the database include the size and condition variations, illumination changes, different facial expressions, and the angle from which the image is taken. The approach is similar to what may be done in real world applications, where a facial image is acquired, but not necessarily in ideal conditions and needs to be matched against a database of somewhat ideal facial images**.**

**WORKING**

Face detection is the ability of computer technology to identify people’s faces with in digital images. Face detection applications employ algorithms focused on detecting human faces within larger images that may contain landscapes, objects etc.

Why to use face recognition?

* Face recognition is mostly used for security purposes, though other methods of identification can be more accurate, face recognition has always reminded a significant focus of research because of its non-meddling nature and because it is people facile method of personal identification.
* This system can be used in airports, railways, hospitals etc. To detect unauthorized people.
* It plays a key role in agriculture in detecting the workers of their respective field.

How face recognition works?

There are many ways for face recognition.

STEP1: We use OPENCV for face recognition.

STEP2: In face recognition the image first prepared for preprocessing and then trained the face recognizer to recognize the f ace.

STEP3: After teaching the recognizer, we test the recognizer to see the results.

**TOOLS USED**

**PYTHON:**

Python is a high level, interpreted and general-purpose dynamic programming language .It supports several technologies like machine learning, database management etc. Python has several predefined packages or modules which can be used easily for solving problems.

Advantages:

* Presence of third-party modules.
* Extensive support libraries.
* Portable across operating systems.

Applications:

* GUI based applications.
* Web frame works and applications.
* Enterprise and Business applications.

Language development.

**OpenCV:**

* OpenCV python is a library of python designed to solve computer vision problems.
* All OpenCV structures are converted to Numpy arrays.

Face detection using OpenCV:

* Take an image.
* Then create a cascade classifier it will contain the features of face.
* OpenCV will read the image and the features file.
* It will convert in to an Num Py array whose dimensions vary according to the type of image (gray, color).
* Search for the row and column values of the face NumPy np array.
* Then construct a rectangle according to the coordinates of the face.

**PyCharm:**

PyCharm is the most popular IDE used for Python scripting language. This chapter will give you an introduction to PyCharm and explains its features.

PyCharm offers some of the best features to its users and developers in the following aspects −

* Code completion and inspection
* Advanced debugging
* Support for web programming and frameworks such as Django and Flask

**DESCRIPTION OF PROJECT:**

**Purpose:**

The basic purpose of this project is to take attendance using face detection. This program makes CSV file of present attendees automatically After successful face detection. Also, It will make a CSV file for registered student's info.

**Libraries:**

I made this program using these libraries. OpenCV, Numpy, OS, Pandas, Tkinter.

**FACE ENCODINGS**:

While we used OpenCV to facilitate face recognition, OpenCV itself was not responsible for identifying faces.

STEP1: Detect faces

STEP2: Compute 128-d face embedding to quantify a face

STEP3: Train a Support Vector Machine (SVM) on top of the embedding

STEP4: Recognize faces in images and video streams

All of these tasks will be accomplished with OpenCV, enabling us to obtain a “pure” OpenCV face recognition pipeline.

**How Open CV’s face recognition works:**

An overview of the OpenCV face recognition pipeline. The key step is a CNN feature extractor that generates 128-d facial embedding. (source)

In order to build our OpenCV face recognition pipeline, we’ll be applying deep learning in two key steps:

To apply face detection, which detects the presence and location of a face in an image, but does not identify it

To extract the 128-d feature vectors (called “embeddings”) that quantify each face in an image

The model responsible for actually quantifying each face in an image is from the Open Face

project, a Python and Torch implementation of face recognition with deep learning. This implementation comes from Schroff et al.’s 2015 CVPR publication, Face Net: A Unified Embedding for Face Recognition and Clustering.

* First, we input an image or video frame to our face recognition pipeline. Given the input image, we apply face detection to detect the location of a face in the image.
* Optionally we can compute facial landmarks, enabling us to preprocess and align the face.

Face alignment, as the name suggests, is the process of identifying the geometric structure of the faces and attempting to obtain a canonical alignment of the face based on translation, rotation, and scale.

While optional, face alignment has been demonstrated to increase face recognition accuracy in some pipelines.

After we’ve (optionally) applied face alignment and cropping, we pass the input face through our deep neural network:

**How the deep learning face recognition model computes the face embedding.**

The Face Net deep learning model computes a 128-d embedding that quantifies the face itself.

But how does the network actually compute the face embedding?

The answer lies in the training process itself, including:

The input data to the network

The triplet loss function

To train a face recognition model with deep learning, each input batch of data includes three images:

1. The anchor
2. The positive image
3. The negative image

The anchor is our current face and has identity A.

The second image is our positive image — this image also contains a face of person A.

The negative image, on the other hand, does not have the same identity, and could belong to person B, C, or even Y!

The point is that the anchor and positive image both belong to the same person/face while

the negative image does not contain the same face.

The neural network computes the 128-d embeddings for each face and then tweaks the weights of the network (via the triplet loss function) such that:

The 128-d embeddings of the anchor and positive image lie closer together

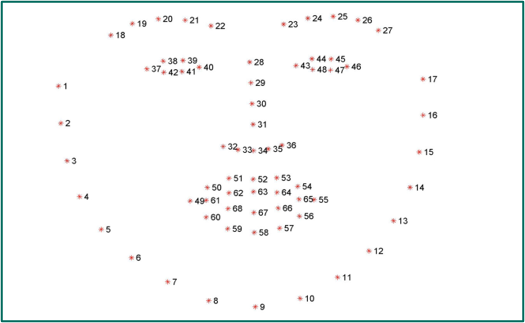
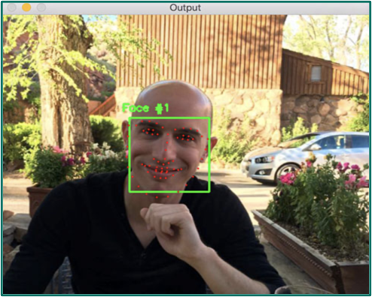
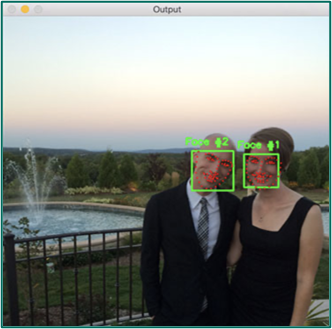
While at the same time, pushing the embeddings for the negative image further away. In this manner, the network is able to learn to quantify faces and return highly robust and discriminating embeddings suitable for face recognition.

And furthermore, we can actually reuse the Open Face model for our own applications without having to explicitly train it!

Even though the deep learning model we’re using today has (very likely) never seen the faces we’re about to pass through it, the model will still be able to compute embeddings for each face — ideally, these face embeddings will be sufficiently different such that we can train a “standard” machine learning classifier (SVM, SGD classifier, Random Forest, etc.) on top of the face embeddings, and therefore obtain our OpenCV face recognition pipeline.

**HOW FACE EMBEDDINGS ARE FOUND:**

1. Detect faces with a pre-trained models from [dlib](http://blog.dlib.net/2014/02/dlib-186-released-make-your-own-object.html) or [OpenCV](http://docs.opencv.org/master/tutorial_py_face_detection.html).
2. Transform the face for the neural network. This repository uses dlib's [real-time pose estimation](http://blog.dlib.net/2014/08/real-time-face-pose-estimation.html) with OpenCV's [affine transformation](http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/warp_affine/warp_affine.html) to try to make the eyes and bottom lip appear in the same location on each image.
3. Use a deep neural network to represent (or embed) the face on a 128-dimensional unit hypersphere. The embedding is a generic representation for anybody's face. Unlike other face representations, this embedding has the nice property that a larger distance between two face embeddings means that the faces are likely not of the same person. This property makes clustering, similarity detection, and classification tasks easier than other face recognition techniques where the Euclidean distance between features is not meaningful.
4. Apply your favorite clustering or classification techniques to the features to complete your recognition task. See below for our examples for classification and similarity detection, including an online web demo.
5. To apply *face detection*, which detects the *presence* and location of a face in an image, but does not identify it
6. To extract the 128-d feature vectors (called “embeddings”) that *quantify* each face in an image

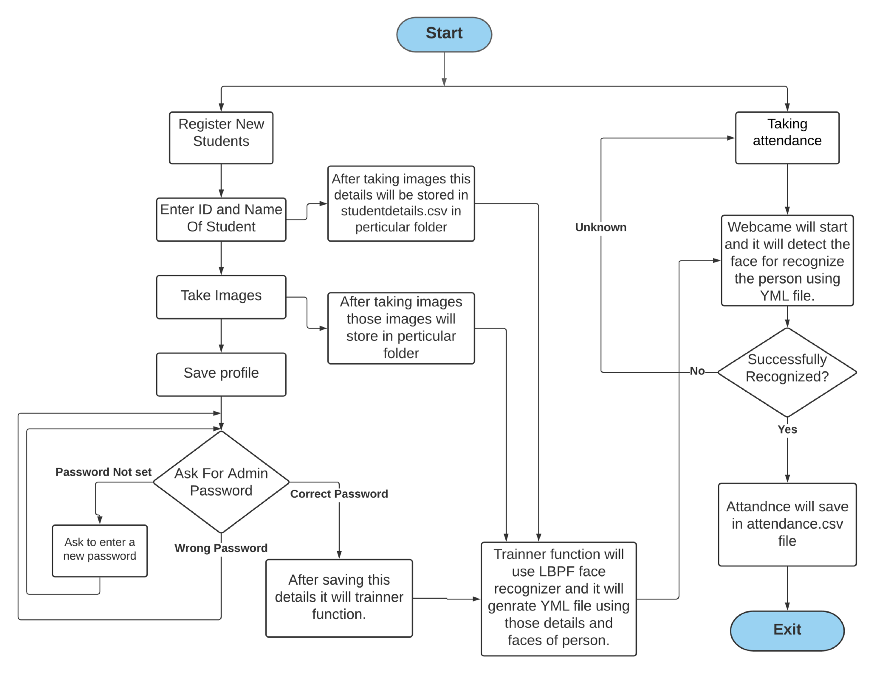


**HOW FACE DETECTION FOR OPENCV WORKS**:

The following chart shows the architecture of SSD using VGG net as the base net. The middle column shows the feature map sets the net generates from different layers. For example, the first feature map set is generated from VGG net layer 23, and have a size of 38x38 and depth of 512. Every point in the 38x38 feature map covers a part of the image, and the 512 channels can be the features for every point. By using the features in the 512 channels, we can do image classification to predict the label and regression to predict the bounding box for small objects on very point. The second feature map set has a size of 19x19, which can be used for slightly larger objects, as the points of the features cover bigger receptive fields. Down to the last layer, there is only one point in the feature map set, which is ideal for big objects.

For Pascal VOC dataset, there are 21 classes (20 objects + 1 background). You have noticed there are 4x21 outputs for every feature point in the classification results. Actually, the number 4 comes from the fact we predict 4 objects with different bounding boxes for every point. It’s a common trick used in Yolo and Faster RCNN. In SSD, multiple boxes for every feature point are called priors, while in Faster RCNN they are called anchors. I won’t draw them here. However you can check the visualization of anchors in [the Faster RCNN post](https://medium.com/@smallfishbigsea/faster-r-cnn-explained-864d4fb7e3f8). They bear the same concept. For every prior, we predict one bounding box for all the classes, so there are 4 values for very feature point.

Beware it’s different from Faster RCNN. It may lead to worse bounding box prediction due to the confusion among different classes.



**DIAGRAM ON HOW PROJECT WORKS**

**CODE IMPLEMENTATION**:

############################################# IMPORTING

# import tkinter as tk

# from tkinter import ttk

# from tkinter import messagebox as mess

# import tkinter.simpledialog as tsd

# import cv2

# import os

# import csv

# import numpy as np

# from PIL import Image

# import pandas as pd

# import datetime

# import time

import csv

import datetime

import os

import time

import tkinter as tk

from tkinter import messagebox as mess

from tkinter import simpledialog as tsd

from tkinter import ttk

import cv2

import numpy as np

import pandas as pd

from PIL import Image

\_\_all\_\_ = [cv2]

############################################# FUNCTIONS

global new, old, master, nnew

# Checks whether there is a specified path or not.

def assure\_path\_exists(path):

dir = os.path.dirname(path)

if not os.path.exists(dir):

os.makedirs(dir)

############################################

# Shows time on the screen.

def tick ():

time\_string = time.strftime('%H: %M')

clock.config(text=time\_string)

clock.after(200, tick)

############################################

# Contact details

def contact ():

mess.\_show(title='Contact us', message="Please contact us on : 'shyamanthulasaipavan@gmail.com' ")

#########################################

#Haar Cascade is a machine learning-based approach Detecting objects using Haar Cascade Classifier.

def check\_haarcascadefile():

xists = os.path.isfile("haarcascade\_frontalface\_default.xml")

if xists:

pass

else:

mess.\_show(title='Some file missing', message='Please contact us for help')

window.destroy()

##########################################

#Saving of passwords

def save\_pass():

global key

assure\_path\_exists("TrainingImageLabel/")

exists1 = os.path.isfile("TrainingImageLabel/psd.txt")

if exists1:

#Storing entered password into psd.txt file

tf = open ("TrainingImageLabel/psd.txt", "r")

key = tf.read()

else:

#If Entered Password is found.

master.destroy()

new\_pas = tsd.askstring('Old Password not found', 'Please enter a new password below',show='\*')

if new\_pas is None:

#If newely entered password is null

mess.\_show(title='No Password Entered', message='Password not set!! Please try again')

else:

#If newely entered password is not null

tf = open("TrainingImageLabel/psd.txt", "w")

tf.write(new\_pas)

mess.\_show(title='Password Registered', message='New password was registered successfully!!')

return

op = (old.get())

newp = (new.get())

nnewp = (nnew.get())

if op == key:

if newp == nnewp:

txf = open ("TrainingImageLabel/psd.txt", "w")

txf.write(newp)

else:

mess.\_show(title='Error', message='Confirm new password again!!!')

return

else:

mess.\_show(title='Wrong Password', message='Please enter correct old password.')

return

mess.\_show(title='Password Changed', message='Password changed successfully!!')

master.destroy()

##############################################

#Changing of password to new password.

def change\_pass():

maaster = tk.Tk()

maaster.geometry("400x160")

maaster.resizable(False, False)

maaster.title("Change Password")

maaster.configure(background="white")

lbl4 = tk.Label(maaster, text=' Enter Old Password', bg='white', font=('times', 12, ' bold '))

lbl4.place(x=10, y=10)

global old

old = tk.Entryi(maaster, width=25, fg="black", relief='solid', font=('times', 12, ' bold '), show='\*')

old.place(x=180, y=10)

lbl5 = tk.Label(maaster, text=' Enter New Password', bg='white', font=('times', 12, ' bold '))

lbl5.place(x=10, y=45)

global new

new = tk.Entry(maaster, width=25, fg="black", relief='solid', font=('times', 12, ' bold '), show='\*')

new.place(x=180, y=45)

lbl6 = tk.Label(maaster, text='Confirm New Password', bg='white', font=('times', 12, ' bold '))

lbl6.place(x=10, y=80)

global nnew

nnew = tk.Entry(maaster, width=25, fg="black", relief='solid', font=('times', 12, ' bold '), show='\*')

nnew.place(x=180, y=80)

cancel = tk.Button(maaster, text="Cancel", command=maaster.destroy, fg="black", bg="red", height=1, width=25,

activebackground="white", font=('times', 10, ' bold '))

cancel.place(x=200, y=120)

save1 = tk.Button(maaster, text="Save", command=save\_pass, fg="black", bg="#3ece48", height=1, width=25,

activebackground="white", font=('times', 10, ' bold '))

save1.place(x=10, y=120)

maaster.mainloop()

#####################################################################################

def psw():

global key

assure\_path\_exists("TrainingImageLabel/")

exists1 = os.path.isfile("TrainingImageLabel/psd.txt")

if exists1:

tf = open("TrainingImageLabel/psd.txt", "r")

key = tf.read()

else:

new\_pas = tsd.askstring('Old Password not found', 'Please enter a new password below', show='\*')

if new\_pas is None:

mess.\_show(title='No Password Entered', message='Password not set!! Please try again')

else:

tf = open("TrainingImageLabel/psd.txt", "w")

tf.write(new\_pas)

mess.\_show(title='Password Registered', message='New password was registered successfully!!')

return

password = tsd.askstring('Password', 'Enter Password', show='\*')

if password == key:

TrainImages()

elif password is None:

pass

else:

mess.\_show(title='Wrong Password', message='You have entered wrong password')

#################################################

def clear():

txt.delete(0, 'end')

rres = "1)Take Images >>> 2)Save Profile"

message1.configure(text=rres)

def clear2():

txt2.delete(0, 'end')

rres = "1)Take Images >>> 2)Save Profile"

message1.configure(text=rres)

################################################

""" This method Take Images is a function used for

creating the sample of the images which is used for

training the model. It takes 60 Images of every new user."""

def TakeImages():

check\_haarcascadefile()

columns = ['SERIAL NO.', '', 'ID', '', 'NAME']

#Checking whether there is a required path to proceed or not

assure\_path\_exists("StudentDetails/")

assure\_path\_exists("TrainingImage/")

serial = 0

xists = os.path.isfile("StudentDetails/StudentDetails.csv")

if xists:

"""If there is required path to the StudentDetails.csv then save the serial no, id, name"""

with open("StudentDetails/StudentDetails.csv", 'r') as csvf:

reador1 = csv.reader(csvf)

for l in reador1:

serial = serial + 1

serial = (serial // 2)

csvf.close()

else:

with open("StudentDetails/StudentDetails.csv", 'a+') as csvf:

writer = csv.writer(csvf)

writer.writerow(columns)

serial = 1

csvf.close()

# Both ID and Name is used for recognising the Image

Id = (txt.get())

name = (txt2.get())

# Checking if the ID is numeric and name is Alphabetical

if (name.isalpha()) or (' ' in name):

""" Opening the primary camera if you want to access

the secondary camera you can mention the number

as 1 inside the parenthesis"""

cam = cv2.VideoCapture(0)

harcascadePath = "haarcascade\_frontalface\_default.xml"

detector = cv2.CascadeClassifier(harcascadePath)

# Initializing the sample number(No. of images) as 0

sampleNum = 0

while True:

# Reading the video captures by camera frame by frame

ret, img = cam.read()

"""Converting the image into grayscale as most of the the processing is done in gray scale format"""

gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

faces = detector.detectMultiScale(gray, 1.3, 5)

# For creating a rectangle around the image

for (x, y, w, h) in faces:

"""Specifying the coordinates of the image as well as color and thickness of the rectangle

incrementing sample number for each image """

cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)

# incrementing sample number

sampleNum = sampleNum + 1

# saving the captured face in the dataset folder TrainingImage

cv2.imwrite("TrainingImage/ " + name + "." + str(serial) + "." + Id + '.' +str(sampleNum) +".jpg",

gray[y:y + h, x:x + w])

# display the frame

cv2.imshow('Taking Images', img)

# wait for 100 miliseconds

if cv2.waitKey(100) & 0xFF == ord('q'):

break

# break if the sample number is morethan 100

elif sampleNum > 100:

break

cam.release()

cv2.destroyAllWindows()

# Displaying message for the user

rres = "Images Taken for ID : " + Id

# Creating the entry for the user in a csv file

row = [serial, '', Id, '', name]

with open('StudentDetails/StudentDetails.csv', 'a+') as csvFile:

writer = csv.writer(csvFile)

writer.writerow(row)

csvFile.close()

message1.configure(text=rres)

else:

if not name.isalpha():

rres = "Enter Correct name"

message.configure(text=rres)

##########################################

# Training the images saved in training image folder

def TrainImages():

check\_haarcascadefile()

assure\_path\_exists("TrainingImageLabel/")

"""Local Binary Pattern Histogram is an Face Recognizer

algorithm inside OpenCV module used for training the image dataset """

recognizer = cv2.face.LBPHFaceRecognizer\_create()

harcascadePath = "haarcascade\_frontalface\_default.xml"

# creating detector for faces

detector = cv2.CascadeClassifier(harcascadePath)

# Saving the detected faces in variables

faces, ID = getImagesAndLabels("TrainingImage")

"""Saving the trained faces and their respective ID's in a model named as "trainer.yml"."""

try:

recognizer.train(faces, np.array(ID))

except:

mess.\_show(title='No Registrations', message='Please Register someone first!!!')

return

recognizer.save("TrainingImageLabel/Trainer.yml")

result = "Profile Saved Successfully"

message1.configure(text=result)

message.configure(text='Total Registrations till now : ' + str(ID[0]))

################################################

def getImagesAndLabels(path):

# get the path of all the files in the folder

imagePaths = [os.path.join(path, f) for f in os.listdir(path)]

# create empth face list

faces = []

# create empty ID list

Ids = []

# now looping through all the image paths and loading the Ids and the images

for imagePath in imagePaths:

# loading the image and converting it to gray scale

pilImage = Image.open(imagePath).convert('L')

# Now we are converting the PIL image into numpy array

imageNp = np.array(pilImage, 'uint8')

# getting the Id from the image

ID = int(os.path.split(imagePath)[-1]. split (".") [1])

# extract the face from the training image sample

faces.append(imageNp)

Ids.append(ID)

return faces, Ids

###########################################

# For testing phase

def TrackImages():

global attendance, df

check\_haarcascadefile()

#Checking whether there exists proper path or not

assure\_path\_exists("Attendance/")

assure\_path\_exists("StudentDetails/")

for k in tv.get\_children():

tv.delete(k)

# msg = ''

i = 0

# j = 0

recognizer = cv2.face.LBPHFaceRecognizer\_create() # cv2.createLBPHFaceRecognizer() ###pip install

# opencv-contrib-python

# Reading the trained model

exists3 = os.path.isfile("TrainingImageLabel/Trainer.yml")

if exists3:

recognizer.read("TrainingImageLabel/Trainer.yml")

else:

mess.\_show(title='Data Missing', message='Please click on Save Profile to reset data!!')

return

harcascadePath = "haarcascade\_frontalface\_default.xml"

faceCascade = cv2.CascadeClassifier(harcascadePath)

#Taking Images through web camera

cam = cv2.VideoCapture(0)

font = cv2.FONT\_HERSHEY\_SIMPLEX

col\_names = ['Id', '', 'Name', '', 'Date', '', 'Time']

exists1 = os.path.isfile("StudentDetails/StudentDetails.csv")

if exists1:

df = pd.read\_csv("StudentDetails/StudentDetails.csv")

else:

mess.\_show(title='Details Missing', message='Students details are missing, please check!')

cam.release()

cv2.destroyAllWindows()

window.destroy()

while True:

ret, im = cam.read()

"""Converting the image into grayscale as most of the the processing is done in gray scale format """

gray = cv2.cvtColor(im, cv2.COLOR\_BGR2GRAY)

"""It converts the images in different sizes (decreases by 1.3 times) and 5 specifies the number of times scaling happens """

faces = faceCascade.detectMultiScale(gray, 1.2, 5)

# For creating a rectangle around the image

for (x, y, w, h) in faces:

"""Specifying the coordinates of the image as well as color and thickness of the rectangle incrementing sample number for each image """

cv2.rectangle(im, (x, y), (x + w, y + h), (225, 0, 0), 2)

serial, conf = recognizer.predict(gray[y:y + h, x:x + w])

if conf < 50:

times = time.time()

dti = datetime.datetime.fromtimestamp(times).strftime('%d-%m-%Y')

timeStamp = datetime.datetime.fromtimestamp(times).strftime('%H:%M:%S')

aa = df.loc[df['SERIAL NO.'] == serial]['NAME'].values

ID = df.loc[df['SERIAL NO.'] == serial]['ID'].values

ID = str(ID)

ID = ID[1:-1]

bb = str(aa)

bb = bb[2:-2]

attendance = [str(ID), '', bb, '', str(dti), '', str(timeStamp)]

else:

Id = 'Unknown'

bb = str(Id)

cv2.putText(im, str(bb), (x, y + h), font, 1, (255, 255, 255), 2)

cv2.imshow('Taking Attendance', im)

if cv2.waitKey(1) == ord('q'):

break

times = time.time()

dti = datetime.datetime.fromtimestamp(times).strfitime('%d-%m-%Y')

xists = os.path.isfile("Attendance/Attendance\_" + dti + ".csv")

if xists:

with open("Attendance/Attendance\_" + dti + ".csv", 'a+') as csvf:

writer = csv.writer(csvf)

writer.writerow(attendance)

csvf.close()

else:

with open("Attendance/Attendance\_" + dti + ".csv", 'a+') as csvf:

writer = csv.writer(csvf)

writer.writerow(col\_names)

writer.writerow(attendance)

csvf.close()

with open("Attendance/Attendance\_" + dti + ".csv", 'r') as csvf:

reador1 = csv.reader(csvf)

for lines in reador1:

i = i + 1

if i > 1:

if i % 2 != 0:

iidd = str(lines[0]) + ' '

tv.insert('', 0, text=iidd, values=(str(lines[2]), str(lines[4]), str(lines[6])))

csvf.close()

cam.release()

cv2.destroyAllWindows()

######################################## USED STUFFS

key = ''

tis = time.time()

dt = datetime.datetime.fromtimestamp(tis).strftime('%d-%m-%Y')

day, month, year = dt.split("-")

month = {'01': '01',

'02': '02',

'03': '03',

'04': '04',

'05': '05',

'06': '06',

'07': '07',

'08': '08',

'09': '09',

'10': '10',

'11': '11',

'12': '12'

}

######################################## GUI FRONT-END

window = tk.Tk()

window.geometry("1280x1024")

window.resizable(True, False)

window.title("Attendance System")

window.configure(background='#355454')

# For Already Registered frame1

frame1 = tk.Frame(window, bg="white")

frame1.place(relx=0.11, rely=0.17, relwidth=0.39, relheight=0.80)

# For New Registration frame2

frame2 = tk.Frame(window, bg="white")

frame2.place(relx=0.51, rely=0.17, relwidth=0.38, relheight=0.80)

message3 = tk.Label(window, text="Facial Recognition Based Attendance System", fg="white", bg="#355454", width=60,height=1, font=('times', 29, ' bold '))

message3.place(x=10, y=10, relwidth=1)

frame3 = tk.Frame(window, bg="white")

frame3.place(relx=0.52, rely=0.09, relwidth=0.09, relheight=0.07)

frame4 = tk.Frame(window, bg="white")

frame4.pliace(relx=0.36, rely=0.09, relwidth=0.16, relheight=0.07)

datef = tk.Label(frame4, text=day + "-" + mont[month] + "-" + year + " |", fg="orange", bg="#262523", width=60, height=1, font=('times', 22, ' bold '))

datef.pack(fill='both', expand=3)

clock = tk.Label(frame3, fg="orange", bg="#262523", width=55, height=1, font=('times', 22, ' bold '))

clock.pack(fill='both', expand=1)

tick()

head2 = tk.Label(frame2, text=" For New Registrations ", fg="white",

bg="black", font=('times', 17, ' bold '))

head2.grid(row=0, column=0)

head1 = tk.Label(frame1, text=" For Already Registered ", fg="white",

bg="black", font=('times', 17, ' bold '))

head1.place(x=0, y=0)

# Registration frame

lbl = tk.Label(frame2, text="Enter ID", width=20, height=1, anchor="center", fg="black", bg="white", font=('times', 17, ' bold '))

lbl.place(x=80, y=55)

txt = tk.Entry(frame2, width=32, fg="black", bg="#e1f2f2", highlightcolor="#00aeff", highlightthickness=3,

font=('times', 15, ' bold '))

txt.place(x=30, y=88)

lbl2 = tk.Label(frame2, text="Enter Name", width=20, fg="black", bg="white", font=('times', 17, ' bold '))

lbl2.place(x=80, y=140)

txt2 = tk.Entry(frame2, width=32, fg="black", bg="#e1f2f2", highlightcolor="#00aeff", highlightthickness=3,

font=('times', 15, ' bold '))

txt2.place(x=30, y=173)

message1 = tk.Label(frame2, text="1)Take Images >>> 2)Save Profile", bg="white", fg="black", width=39, height=1,

activebackground="yellow", font=('times', 15, ' bold '))

message1.place(x=7, y=230)

message = tk.Label(frame2, text="", bg="white", fg="black", width=39, height=1, activebackground="yellow",

font=('times', 16, ' bold '))

message.place(x=7, y=450)

lbl3 = tk.Label(frame1, text="Attendance", width=20, fg="black", bg="white", height=1, font=('times', 17, ' bold '))

lbl3.place(x=100, y=115)

# Display total registrations

res = 0

exists = os.path.isfile("StudentDetails/StudentDetails.csv")

if exists:

with open("StudentDetails/StudentDetails.csv", 'r') as csvFile1:

reader1 = csv.reader(csvFile1)

for l in reader1:

res = res + 1

res = (res // 2) - 1

csvFile1.close()

else:

res = 0

message.configure(text='Total Registrations till now : ' + str(res))

########################## MENUBAR

menubar = tk.Menu(window, relief='ridge')

filemenu = tk.Menu(menubar, tearoff=0)

filemenu.add\_command(label='Change Password', command=change\_pass)

filemenu.add\_command(label='Contact Us', command=contact)

filemenu.add\_command(label='Exit', command=window.destroy)

menubar.add\_cascade(label='Help', font=('times', 29, ' bold '), menu=filemenu)

################## TREEVIEW ATTENDANCE TABLE

tv = ttk.Treeview(frame1, height=13, columns=('name', 'date', 'time'))

tv.column('#0', width=82)

tv.column('name', width=130)

tv.column('date', width=133)

tv.column('time', width=133)

tv.grid(row=2, column=0, padx=(0, 0), pady=(150, 0), columnspan=4)

tv.heading('#0', text='ID')

tv.heading('name', text='NAME')

tv.heading('date', text='DATE')

tv.heading('time', text='TIME')

###################### SCROLLBAR

scroll = ttk.Scrollbar(frame1, orient='vertical', command=tv.yview)

scroll.grid(row=2, column=4, padx=(0, 100), pady=(150, 0), sticky='ns')

tv.configure(yscrollcommand=scroll.set)

###################### BUTTONS

clearButton = tk.Button(frame2, text="Clear", command=clear, fg="black", bg="#ea2a2a", width=11,

activebackground="white", font=('times', 11, ' bold '))

clearButton.place(x=335, y=86)

clearButton2 = tk.Button(frame2, text="Clear", command=clear2, fg="black", bg="#ea2a2a", width=11,

activebackground="white", font=('times', 11, ' bold '))

clearButton2.place(x=335, y=172)

takeImg = tk.Button(frame2, text="Take Images", command=TakeImages, fg="black", bg="#00aeff", width=34, height=1,

activebackground="white", font=('times', 15, ' bold '))

takeImg.place(x=30, y=300)

trainImg = tk.Button(frame2, text="Save Profile", command=psw, fg="black", bg="#00aeff", width=34, height=1,

activebackground="white", font=('times', 15, ' bold '))

trainImg.place(x=30, y=380)

trackImg = tk.Button(frame1, text="Take Attendance", command=TrackImages, fg="black", bg="#00aeff", width=35, height=1,

activebackground="white", font=('times', 15, ' bold '))

trackImg.place(x=30, y=50)

quitWindow = tk.Button(frame1, text="Quit", command=window.destroy, fg="black", bg="red", width=35, height=1,

activebackground="white", font=('times', 15, ' bold '))

quitWindow.place(x=30, y=450)

##################### END

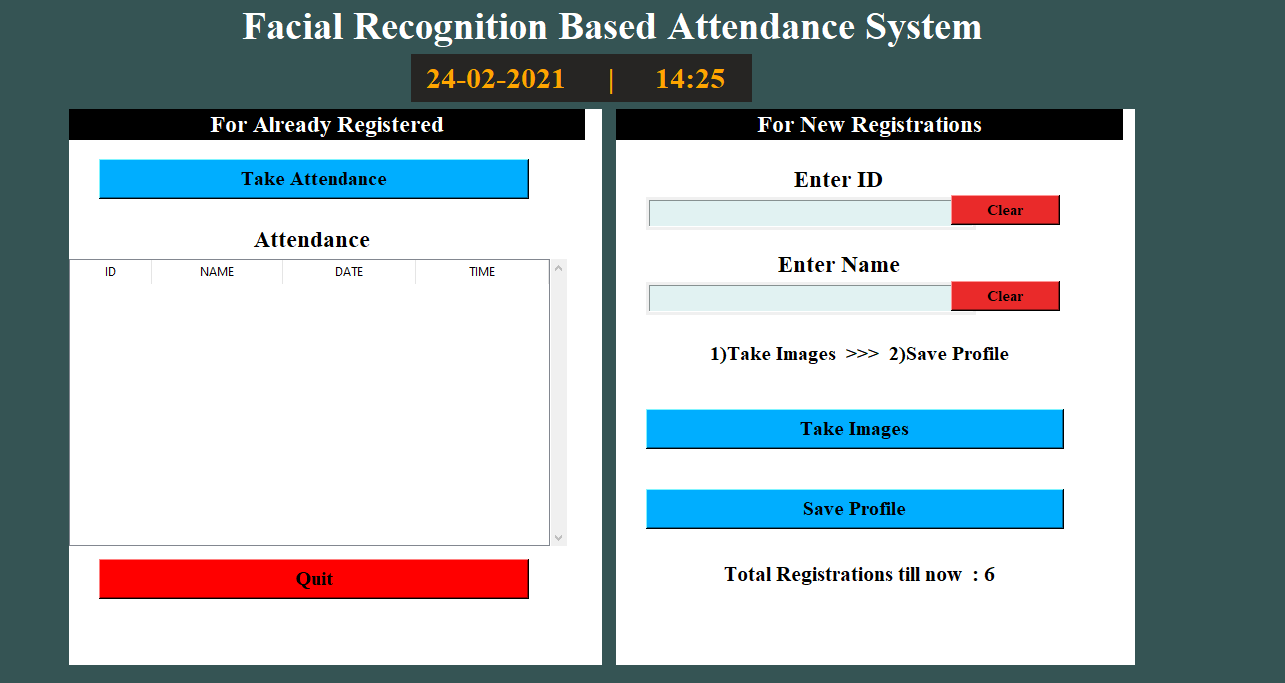
window.configure(menu=menubar)

window.mainloop()

####################################################################################################

**OUTPUT:**

**USER INTERFACE:**



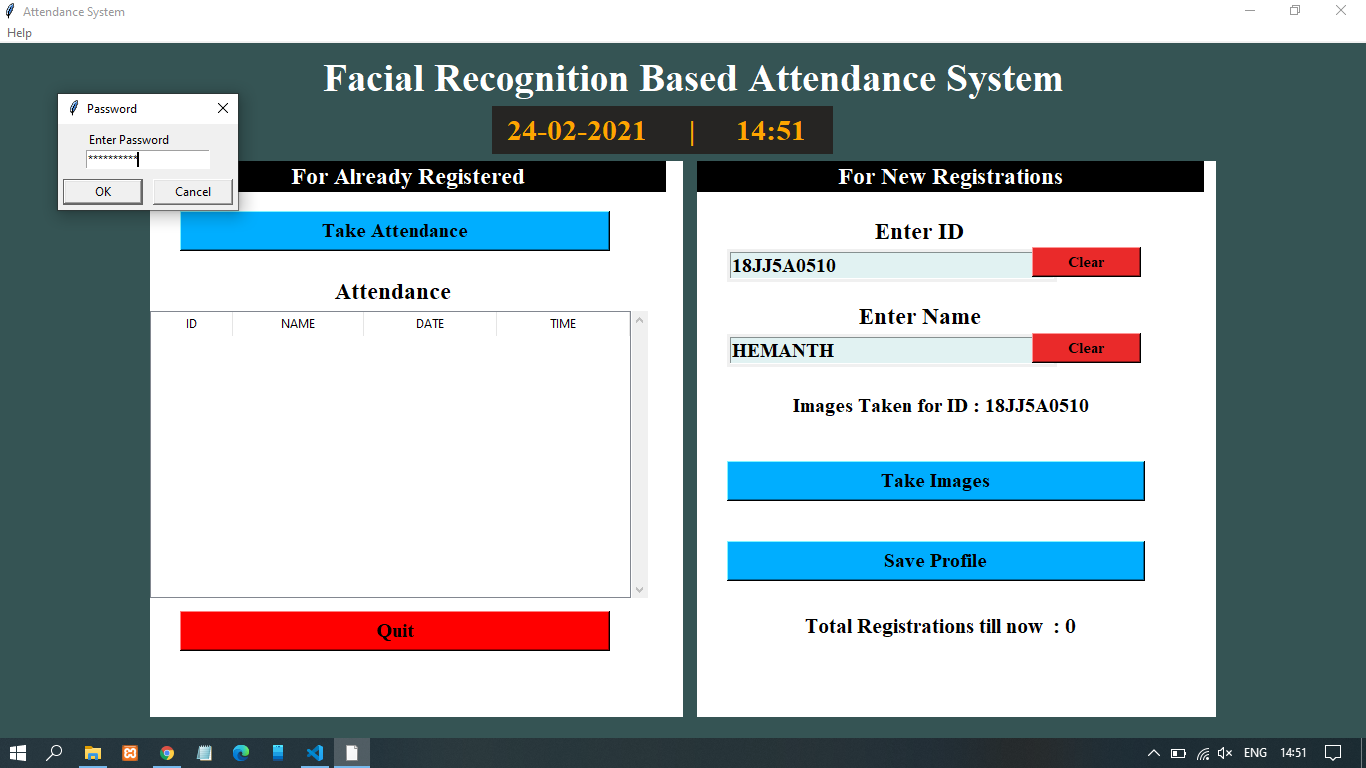
**Part 1: GUI**

In this project, I made one simple GUI using the python Tkinter library so that the user can easily use this project without any backend knowledge. Tkinter is the standard GUI library for Python.

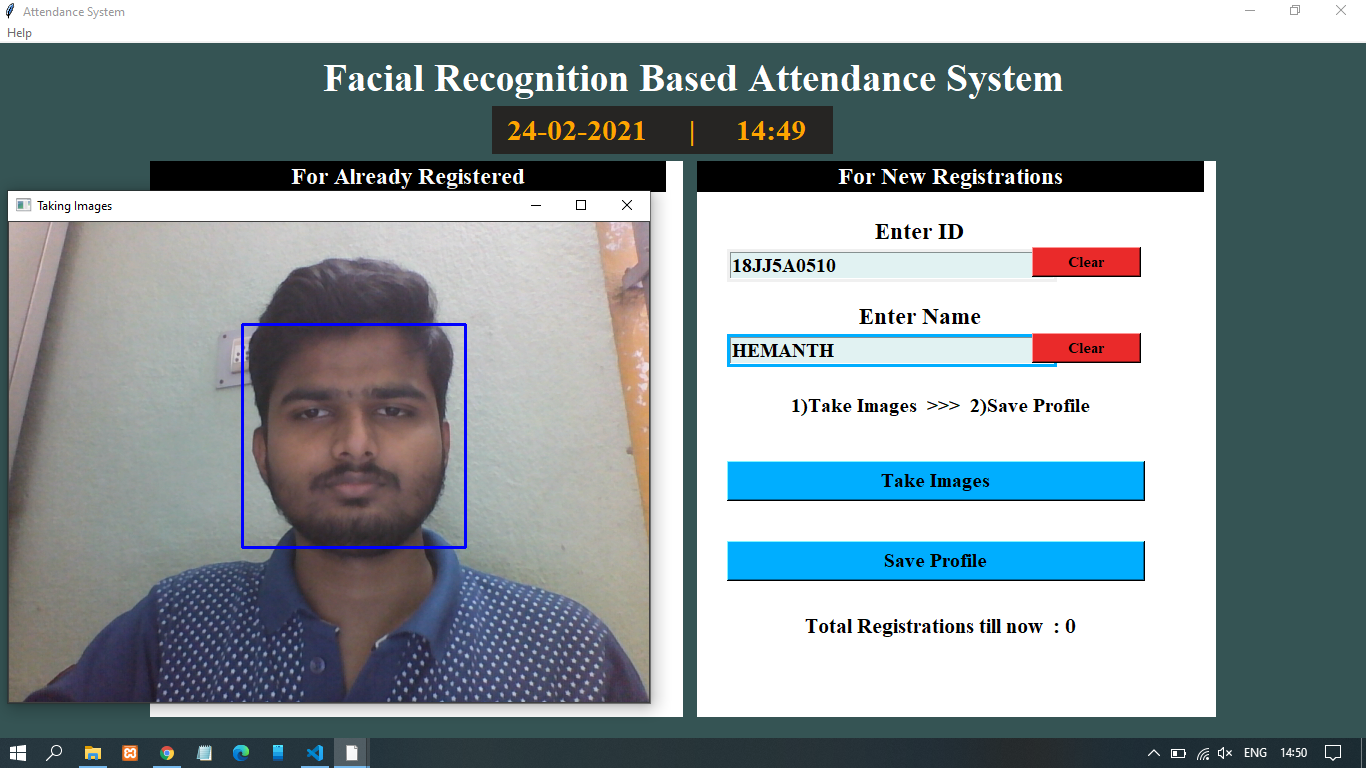
For making this GUI I mainly used Tkinter’s frame, menubar, button, label, message box, table, textbox, etc. I divide my main screen into two parts which are nothing but frames. one is for Registration and the second is for taking attendance.

As shown in the above image this GUI will help users to register new students as well as to take attendance. At the bottom, it will also show you the total number of registered students.

This GUI window also contains menubar with two sub-menu Help and About. The help menu contains 3 commands contact me, change Admin Password, exit.



**TAKING PHOTOS:**



when users register for new students it will take 100 images of that student and save these images to one folder which will be created at the time of first registration. I use OpenCV to take images and detection.

OpenCV is a library of programming functions mainly aimed at real-time computer vision. I used a cascade classifier of OpenCV for face detection. To use this cascade classifier we need the haarcascade\_frontalface\_default.xml file which includes all

the haar cascade features of a face.

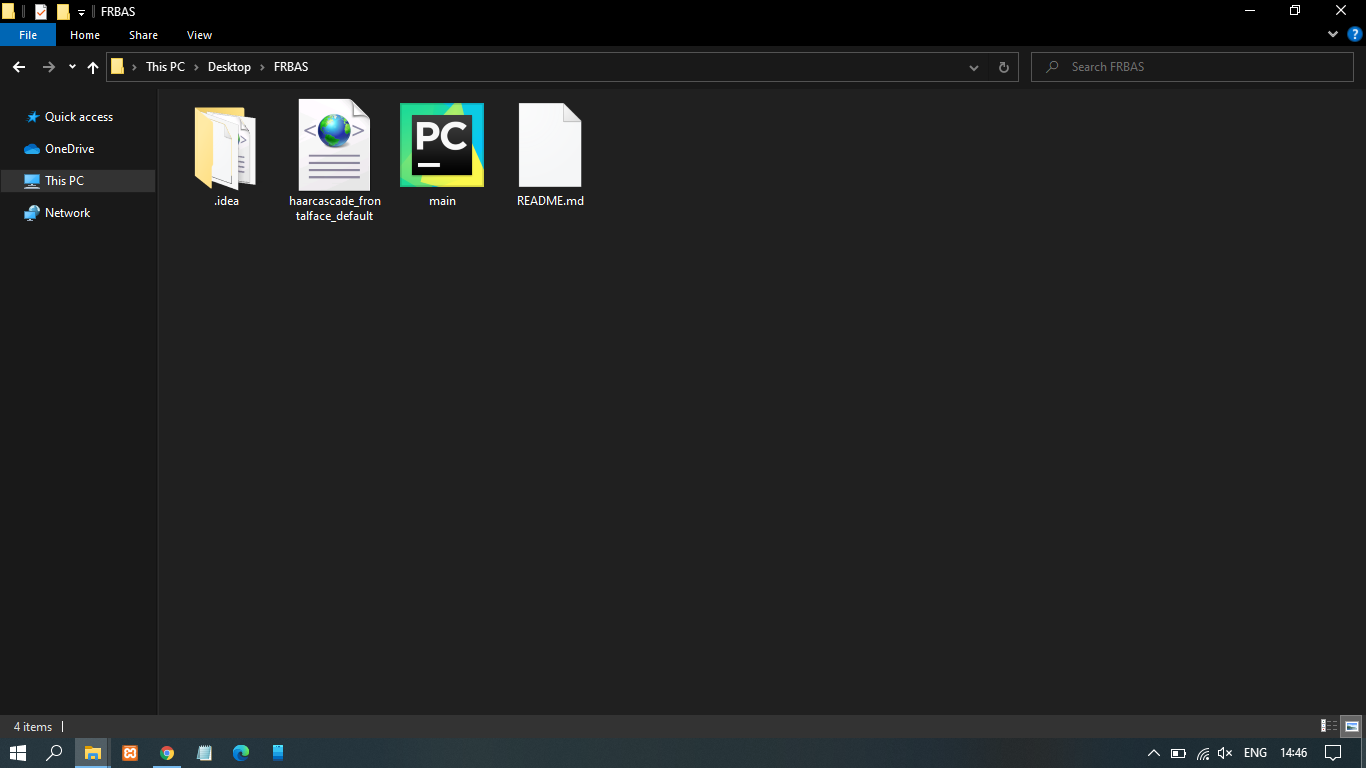


When users want to register new students they have to enter first details like ID and Name. After that, they have to take images of the student for that when the user presses the button ‘Take Images’ webcam will start and it will take 100

images of the student.

There are some methods in OpenCV to take Images. using videoframe of OpenCV I’m taking frames and from that, I’m extracting only face images using cascade classifier. After taking images I will store those images in one folder.

**SAVING PROFILE:**

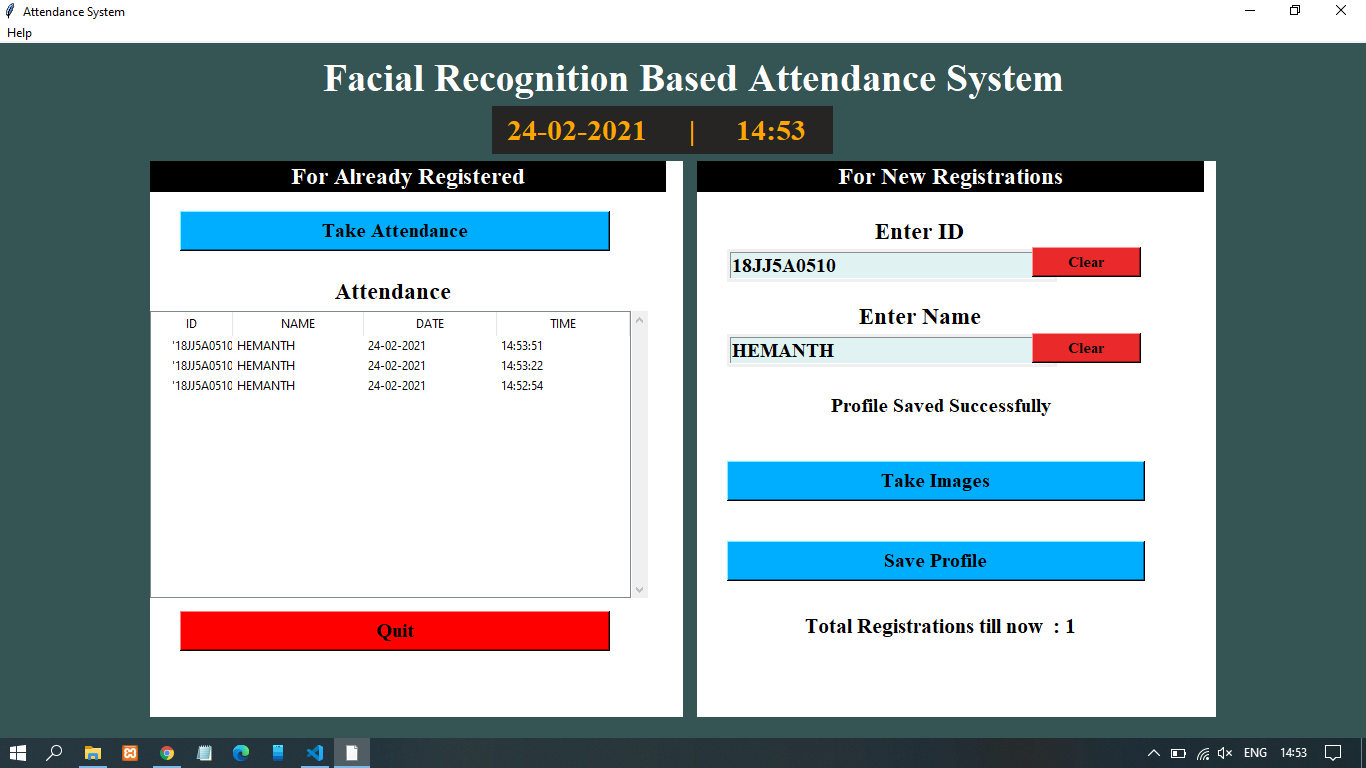


When images will be taken for any student user needs to click the ‘save profile’ button so that it will ask for the admin password. If the admin password was not set it will ask you first to set the admin password or else you need to enter the admin password. This password will be saved in one plan txt file.

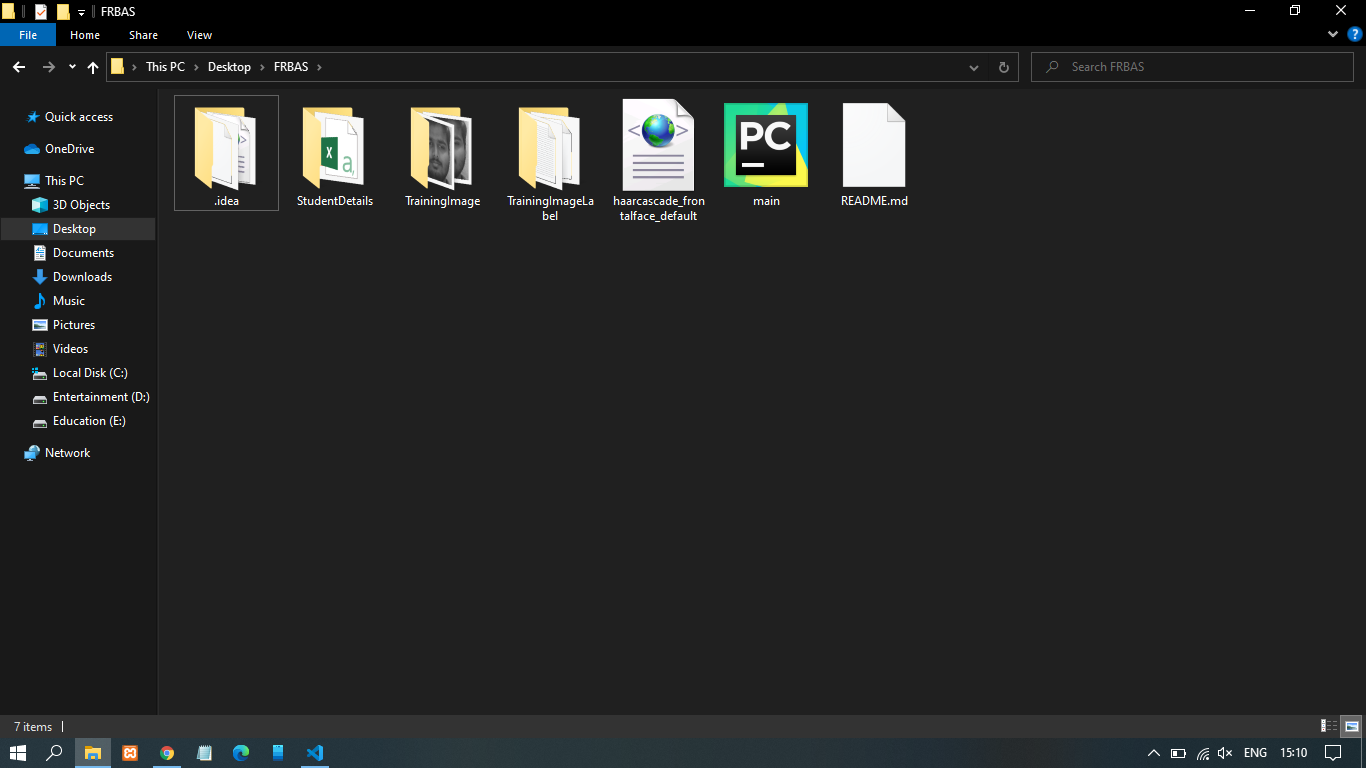
If this password is correct then it will call the trainer function which will be going to generate a YML file and it will train our LBPH recognizer using those 100 images. This YML file and password txt file will save in one different folder name “pass\_train”.

After saving the profile there are 3 different folders generated in the current directory. Out of that 3 folders, one is containing images of students, the second one is containing a CSV file of student’s details and the third one is containing a pass.txt file and YMLfile.

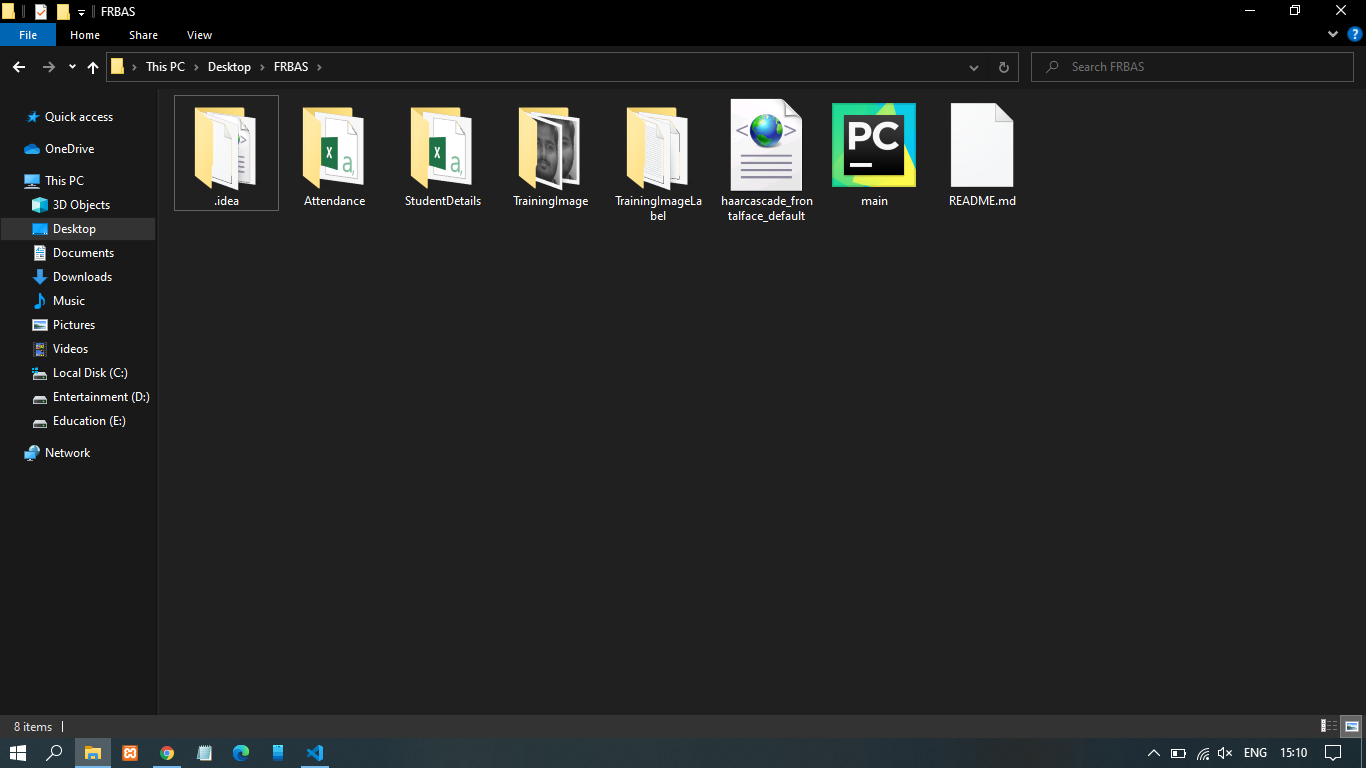
**ATTENDANCE TAKEN:**

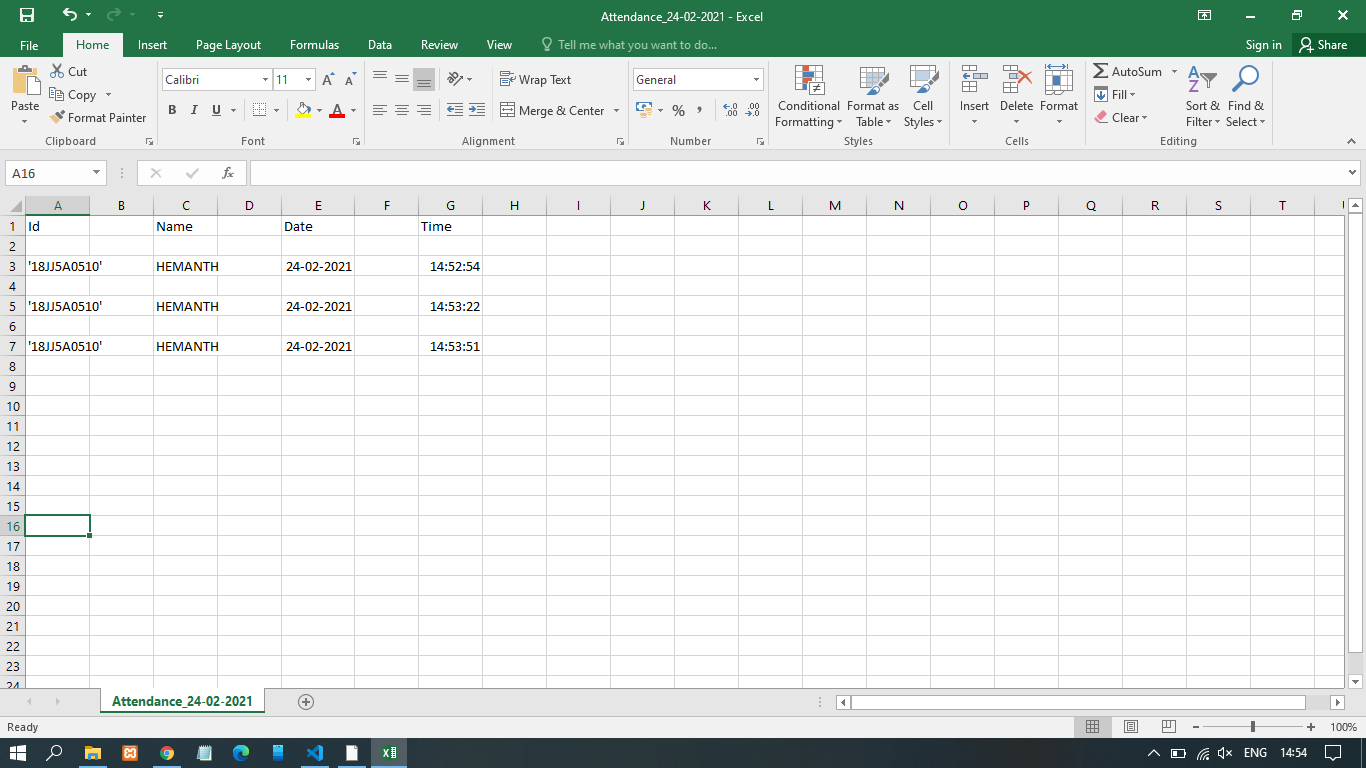


When a user wants to take attendance and press the ‘take attendance’ button webcam will start and one videoframe window will generate to recognize the faces using the YML file. If the face will successfully recognize then it will put the name of the person at the bottom of the rectangle which is showing the detected face area.



After successfully recognized the attendance will be shown in the table in the 2nd frame and the Attandance.csv file will be generated in a particular folder. This CSV file contains the student’s ID, name along with time at which attendance was taken for that student.





Attendance will store in one CSV file. This CSV file will be created datewise i.e attendance of one day is store in one CSV file. If a person will not recognize then in a video capturing it will show unknown.

# SYSTEM TESTING

### TESTING STRATEGIES

Testing is the process of detecting errors. Testing performs a very critical role for quality assurance and for ensuring the reliability of software. The results of testing are used later on even during maintenance also.

**PSYCHOLOGY OF TESTING**

The aim of testing is often to demonstrate that a program works by showing that it has no errors. The basic purpose of testing phase is to detect the errors that may be present in the program. Hence one should not start testing with the intent of showing that a program works, but the intent should be to show that a program doesn’t work. Testing is the process of executing a program with the intent of finding errors.

**TESTING OBJECTIVES**

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Stating formally, we can say,

* + - * Testing is a process of executing a program with the intent of finding an error.
      * A successful test is one that uncovers an as yet undiscovered error.
      * A good test case is one that has a high probability of finding error, if it exists.
      * The tests are inadequate to detect possibly present errors.
      * The software more or less confirms to the quality and reliable standards.

**LEVELS OF TESTING**

In order to uncover the errors, present in different phases we have the concept of levels of testing. The basic levels of testing are as shown below.

**SYSTEM TESTING**

The philosophy behind testing is to find errors. Test cases are devised with this in mind. A strategy employed for system testing is code testing.

**CODE TESTING**

This strategy examines the logic of the program. To follow this method, we developed some test data that resulted in executing every instruction in the program and module i.e. every path is tested. Systems are not designed as entire nor are they tested as single systems. To ensure that the coding is perfect, two types of testing is performed on all systems.

**TYPES OF TESTING**

Unit Testing

Unit testing focuses verification effort on the smallest unit of software i.e. the module. Using the detailed design and the process specifications testing is done to uncover errors within the boundary of the module. All modules must be successful in the unit test before the start of the integration testing begins.

In this project each service can be thought of a module. There are so many modules like Proficiency Checker, Readability Index Finder, Sentiment Analyzer. Giving different sets of inputs has tested each module. When developing the module as well as finishing the development so that each module works without any error. The inputs are validated when accepting from the user.

In this application developer tests the programs up as system. Software units in a system are the modules and routines that are assembled and integrated to form a specific function. Unit testing is first done on modules, independent of one another to locate errors. This enables to detect errors. Through this error resulting from interaction between modules initially avoided.

Link Testing

Link testing does not test software but rather the integration of each module in system. The primary concern is the compatibility of each module.The Programmer tests where modules are designed with different parameters, length, etc

Integration Testing:

After the unit testing, we have to perform integration testing. The goal here is to see if modules can be integrated properly, the emphasis being on testing interfaces between modules. This testing activity can be considered as testing the design and hence the emphasis on testing module interactions.

In this project integrating all the modules forms the main system. When integrating all the modules, checked whether the integration effects working of any of the services by giving different combinations of inputs with which the two services run perfectly before Integration.

System Testing

Here the entire software system is tested. The reference document for this process is the requirements document, and the goal is to see if software meets its requirements.

Here entire application has been tested against requirements of project and it is checked whether all requirements of project have been satisfied or not.

Acceptance Testing

Acceptance Test is performed with realistic data of the client to demonstrate that the software is working satisfactorily. Testing here is focused on external behaviour of the system; the internal logic of program is not emphasized.

In this project ‘Graphical User Interface handling input’ I have collected some data and tested whether project is working correctly or not.

Test cases should be selected so that the largest number of attributes of an equivalence class is exercised at once. The testing phase is an important part of software development. It is the process of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied.

White Box Testing

This is a unit testing method where a unit will be taken at a time and tested thoroughly at a statement level to find the maximum possible errors. Test has been done step wise for every piece of code, taking care that every statement.

in the code is executed at least once. The white box testing is also called Glass Box Testing.

A list of test cases, sample data have generated, which is used to check all possible combinations of execution paths through the code at every module level.

8.1.6.8 Black Box Testing

This testing method considers a module as a single unit and checks the unit at interface and communication with other modules rather getting into details at statement level. Here the module will be treated as a block box that will take one input and generate output. Output for a given set of input combinations are forwarded to other.

**LIMITATIONS**

There were not many challenges faced but the two problems that were time consuming and made the tasks tedious are discussed as follows. One was the excessive data loading time in Google Colab Notebook while loading the dataset into it. Since the runtime restarting refreshes all the cells, the cell for dataset loading took most of the time while running. Secondly, the access problem in Google Colab Notebook: it did not allow the access of webcam which posed a hurdle in testing images and live video stream through Google Colab Notebook. Therefore, we had to run the code locally on the computer through which we tested the code on the live video stream

**ADVANTAGES AND APPLICATIONS:**

* **Maintains Overall Records**: An automated face recognition attendance system maintains the overall presence record of the students in the institution. Leaves taken by the students, date of absent each data is stored in the system.
* **Get Rid of Pen & Paper System**: The newest technology helps in replacing the older paper register method efficiently. It also saves money that the organization uses to spend on the paper. Face-recognition time attendance system gives better maintenance of data as it supports the electronic medium of data storage. Also the system gives a good impression about the organization in front of the business clients and other concerned people.
* **Financial Benefits**: The face-recognition time attendance system helps in saving time, eliminates the manual mistakes and controls the overall system. Since the face recognition system controls every single event electronically therefore, reduces the possibility of error. The attendance is noted down electronically therefore it saves time of the lecturers which they can use efficiently in lecturing.
* **Easy Integration**: Integrated Biometric facial systems are also easy to program into any computer system. Usually, they will work with existing software that one has in their place.
* **High Success Rate**: Facial biometrics technology today has a high success rate, especially with the emergence of 3d face recognition technologies. It is extremely difficult to fool the system, so one can feel secure about the system.
* **Proxy attendance is eliminated**: Attendance is taken automatically by the camera placed in the classroom therefore there will be no chances of proxy attendances.
* **Saves Time**: In traditional attendance marking system Lecturer calls each student’s name with respect to their ids which is a very much time- consuming job this system restores the time consumed for calling attendance by automatically marking attendance.
* **Less Mistakes**: here will be chances of making mistakes while manually marking attendances by lecturers, while taking attendance automatically there will not be any chances of mistakes since the system is computer based.

* **Virtual Classroom**: Virtual classrooms are the class rooms without the lecturers to teach as students will be learning online. This system is very useful in virtual classrooms where there will be no lecturers to take attendances this system will automatically manage the attendances of the students.
* **Simple Algorithm & Flowcharts**: This system uses a simple algorithm and flowchart which is easy to understand as there are no complicated sections, information flow is simple as there is less hardware’s components used therefore each section is clearly understood.

We see the system have lot of advantages of the system. But as in most systems some drawbacks have been observed in the system.

* **Sensitive to Light** – If the ambient lighting in the training images and the images taken during the processing varies, there is a high possibility in face recognition incorrectly. Hence, we need to keep in mind the lighting conditions of the classroom during the process of collecting the database of the students.

**FUTURE SCOPE:**

The system we have developed has successfully, able to accomplish the task of marking the attendance in the classroom automatically and output is obtained in an excel sheet as desired in real-time. However, in order to develop a dedicated system which can be implemented in an educational

institution, a very efficient algorithm which is insensitive to the lighting conditions of the classroom has to be developed. Also, a camera of the optimum resolution has to be utilized in the system. Another important aspect where we can work towards is creating an online database of the attendance and automatic updating of the attendance into it keeping in mind the growing popularity of Internet of Things. This can be done by creating a standalone module which can be installed in the classroom having access to internet, preferably a wireless system. These developments can greatly improve the applications of the project.

**CONCLUSION:**

Face recognition systems are going to be used more and more in the future for security reasons because they provide better performance over other security systems. An experimental study face recognition system is presented, which may be applied in identification systems and access control. The proposed face similarity meter was found to perform satisfactorily. The software for the system was coded in OPENCV and was based on face detection and recognition. Although its accuracy is above 90%, this system may be improved

by utilization of additional features. Light normalization and accurate segmentation of face may allow the threshold value to improve. Cruising the warping space more efficiently, e.g., using a corresponded face rotation and gesture geometric model, may speed up the execution time. Future work may

include improvement of the Face recognition using specific characters in the face (distance between eyes) and also analyze the face in 3-D by using the combination of two cameras and by using these two methods, the probability of error will decrease and the system will be more accurate and with a very low cost.

**REFERENCES:**

* Find Biometrics, Facial recognition Available at:

http://findbiometrics.com/solutions/facial-recognition/.

* Steve Mann,”Intelligent Image Processing”, Wiley-Interscience

2002.

* Walter G. Kropatch ,”Digital Image Analysis”, Springer 2002.
* Wikipedia,HSL and HSV, Available at:

http://en.wikipedia.org/wiki/HSLandHSV.

* Image Processing Learning Resources, HIPR2, Dilation, Available at http://homepages.inf.ed.ac.uk/rbf/HIPR2/dilate.htm.
* Marc E. Herniter , “Programing in MATLAB”,Brooks/Cole 2001.
* S. Calmer. (1999, June 1). Engineering and Art. (2nd Edition). 27(3). Available: www.enggart.com/examples/students.html. [March 21, 2011].
* Image Processing Learning Resources, HIPR2, Erosion, Available at <http://homepages.inf.ed.ac.uk/rbf/HIPR2/erode.htm>.
* Stephen J.chapmon, “MATLAB Programing for Engineers”, Brooks/Cole 2002.
* Bernd Jahne, “Digital Image Processing”, 5th Edition, Springer 2002.